

P21105.A01

21. (Amended-Clean Text) A method as claimed in claim 13 wherein after said first process finishes executing data, program modules and execution states from said first process are added to said sub-process and said sub-process is reactivated.

REMARKS

By the above amendment, the claims have been amended to delete multiple dependency.

If there should be any questions, the Examiner is invited to contact the undersigned at the telephone number listed below.

Respectfully submitted,
Hwee Hwa PANG and Teow Hin NGAIR

Bruce H. Bernstein
Reg. No. 29,027

June 15, 2001
GREENBLUM & BERNSTEIN, P.L.C.
1941 Roland Clarke Place
Reston, VA 20191
(703) 716-1191

MARKED-UP COPY OF AMENDED CLAIMS

3. (Amended) A method as claimed in claim 1 [or 2] wherein a construct is formed for storing said sub-process, while said first process retains the process identity of said computing process.

6. (Amended) A method as claimed in claim 3 [any of claims 3 to 5] wherein said construct is provided with and authorising signature.

7. (Amended) A method as claimed in claim 3 [any of claims 3 to 6] wherein said construct is sent to a memory storage device.

8. (Amended) A method as claimed in claim 3 [any of claims 3 to 7] wherein said sub-process is a dormant process comprising data, program code and execution states of said computing process temporarily not required by said first process, and wherein said first process is able to re-acquire data, program codes and execution states from said dormant process as and when required by said first process.

10. (Amended) A method as claimed in claim 8 [or 9] wherein said sub-process comprises data, program code and execution states specific to a given user of a computing means, and wherein said second process is formed to temporarily store said data, program code and execution states while said user is absent from said computing means.

11. (Amended) A method as claimed in claim 3 [any of claims 3 to 10] wherein after said first process finishes executing data, program modules and execution states from said first process are added to said sub-process and said sub-process is reactivated.

13. (Amended) A method as claimed in claim 1 [or 2] wherein a construct is formed for storing said sub-process, while said first process is run in place of said computing process.

16. (Amended) A method as claimed in claim 13 [any of claims 13 to 15] wherein said construct is provided with and authorising signature.

17. (Amended) A method as claimed in claim 13 [any of claims 13 to 16] wherein said construct is sent to a memory storage device.

18. (Amended) A method as claimed in claim 13 [any of claims 13 to 17] wherein said sub-process is a dormant process comprising data, program code and execution states of said computing process temporarily not required by said first process, and wherein said first process is able to re-acquire data, program codes and execution states from said dormant process as and when required by said first process.

P21105.A01

20. (Amended) A method as claimed in claim 18 [or 19] wherein said sub-process comprises data, program code and execution states specific to a given user of a computing means, and wherein said second process is formed to temporarily store said data, program code and execution states while said user is absent from said computing means.

21. (Amended) A method as claimed in claim 13 [any of claims 13 to 20] wherein after said first process finishes executing data, program modules and execution states from said first process are added to said sub-process and said sub-process is reactivated.